Advances in Applied Mathematics **25,** 194–211 (2000)
doi:10.1006/aama.1999.0693, available online at http://www.idealibrary.com on IDEAL®

A Combinatorial Theory of Higher-Dimensional Permutation Arrays[1]

Kimmo Eriksson

E-mail: Kimmo.Eriksson@mdh.se

and

Svante Linusson

Department of Mathematics, Stockholm University,
SE-106 91 Stockholm, Sweden
E-mail: linusson@matematik.su.se

Received May 22, 1999; accepted April 5, 2000

We define a class of hypercubic (shape $[n]^d$) arrays that in a certain sense are d-dimensional analogs of permutation matrices with our motivation from algebraic geometry. Various characterizations of permutation arrays are proved, an efficient generation algorithm is given, and enumerative questions are discussed although not settled. There is a partial order on the permutation arrays, specializing to the Bruhat order on S_n when d equals 2, and specializing to the lattice of partitions of a d-set when n equals 2.

Key Words: permutations; high-dimensional; flags; intersections; Bruhat order; partition lattice.

1. INTRODUCTION

The aim of this paper is to develop a purely combinatorial theory of higher-dimensional permutation arrays, with our motivation coming from algebraic geometry. The fundamental connection to geometry is that the

[1] Both authors were partially supported by the NFR and the second author was also supported by MSRI.

0196-8858/00 $35.00

strata of a generalized Bruhat decomposition of products of flag manifolds are indexed by these pemutation arrays. This connection is treated in another report by the same authors [1].

A permutation matrix of size n is a matrix of shape $n \times n$ with exactly one numeral 1 in each row and column, all other entries being 0. For simplicity, we replace the ones by dots and replace the zeros by empty entries to obtain a two-dimensional dot array. For example, the dotted version of

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \text{ is } \begin{array}{|c|c|}\hline & \bullet \\ \hline \bullet & \\ \hline \end{array}.$$

We will suggest a generalization of the concept of the permutation matrix to arbitrary dimensions. By an $[n]^d$-array we shall mean a hypercubic array $n \times \cdots \times n$ of dimension d. For each n we shall present a family of dotted $[n]^d$-arrays, such that for $d = 2$ we obtain the ordinary permutation matrices of size n.

1.1. *Nice Tries* Several generalizations of permutation matrices to higher dimensions can be proposed, each generalizing some combinatorial aspect of the classical case. (We do not at all consider the group theoretic aspects of permutations.)

One natural candidate is the *dense* d-dimensional permutation array, where we have distributed n^{d-1} dots in an $[n]^d$-array, such that there is exactly one dot in each one-dimensional subarray of size n. There are two dense three-dimensional permutation arrays of size 2 (the left (2×2)-matrix denotes the upper layer of the three-dimensional array):

$$\begin{array}{|c|c|}\hline \bullet & \\ \hline & \bullet \\ \hline \end{array}\begin{array}{|c|c|}\hline & \bullet \\ \hline \bullet & \\ \hline \end{array} \quad \text{and} \quad \begin{array}{|c|c|}\hline & \bullet \\ \hline \bullet & \\ \hline \end{array}\begin{array}{|c|c|}\hline \bullet & \\ \hline & \bullet \\ \hline \end{array}.$$

Another candidate is the *sparse* d-dimensional permutation matrix, where n dots are distributed so that we have exactly one dot in each submatrix of size n and codimension one (that is, dimension $d - 1$). There are four sparse three-dimensional permutation arrays of size 2:

$$\begin{array}{|c|c|}\hline \bullet & \\ \hline & \\ \hline \end{array}\begin{array}{|c|c|}\hline & \\ \hline & \bullet \\ \hline \end{array}, \quad \begin{array}{|c|c|}\hline & \\ \hline & \bullet \\ \hline \end{array}\begin{array}{|c|c|}\hline \bullet & \\ \hline & \\ \hline \end{array}, \quad \begin{array}{|c|c|}\hline & \\ \hline \bullet & \\ \hline \end{array}\begin{array}{|c|c|}\hline & \bullet \\ \hline & \\ \hline \end{array}, \quad \begin{array}{|c|c|}\hline & \bullet \\ \hline & \\ \hline \end{array}\begin{array}{|c|c|}\hline & \\ \hline \bullet & \\ \hline \end{array}.$$

A dense three-dimensional array is equivalent to a latin square. Since the combinatorics of latin squares is notoriously difficult, it seems difficult to say much in general about dense arrays.

The sparse d-dimensional arrays, on the other hand, correspond to elements in the product of $d - 1$ copies of the symmetric group S_n, with $(\pi_1, \pi_2, \ldots, \pi_{d-1})$ giving the sparse array with a dot in $(i, \pi_1(i), \pi_2(i), \ldots, \pi_{d-1}(i))$ for each $i = 1, 2, \ldots, n$. Hence, there are $(n!)^{d-1}$ different sparse arrays of size n in dimension d. The sparse arrays were used by Pascal in 1900 to define higher-dimensional determinants [3].

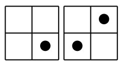
1.2 *Our Version* For the geometric application that we have in mind (i.e., encoding of a notion of relative positions of d complete flags [1]), none of the above suggestions is appropriate. Instead, the following properties of permutation matrices turn out to be the crucial ones:

(i) Every row and column has (at least) one dot.

(ii) In every upper left submatrix, the number of columns containing a dot equals the number of rows containing a dot.

(iii) The set of dots is minimal such that (i) and (ii) are satisfied.

The dense arrays above do not satisfy any correspondingly defined condition. The sparse arrays obviously do, but they are not the only ones. For example, for size 2 and dimension 3, there is also


.

We shall see that this candidate is a generic example of a nonsparse permutation in the sense that we will define in detail in this paper, the outline of which is as follows:

We begin with a series of elementary but necessary definitions, leading up to the concept of *totally rankable* dot arrays. We then prove that there is a canonical representative of any rank equivalence class of totally rankable dot arrays. This canonical representative has a minimal number of dots. *Permutation arrays* are defined as such minimal totally rankable dot arrays. Our results include:

- A characterization theorem of totally rankable dot arrays.
- A characterization of rank arrays of permutation arrays.
- An algorithm for fast generation of all permutation arrays for given values of n and d.
- Some enumerative data obtained by this algorithm.

In our companion paper [1] we investigate the algebraic implications. There we find that two natural partial orders can be defined on permutation arrays, which both coincide with the Bruhat order of S_n for $d = 2$ and are isomorphic to the partition lattice of a d-set for $n = 2$. The combinatorially defined order is presented in Section 6.

2. RANK AND RANK EQUIVALENCE OF DOT ARRAYS

A permutation matrix of size n is an $(n \times n)$-matrix with exactly one dot in each row and column, all other entries being empty. A trivial but useful observation is that the number of dots in a given submatrix is equal to the number of its rows that contain a dot and is also equal to the number of its columns that contain a dot. By simple linear algebra, the number of dots in a submatrix equals its matrix rank if the dots denote ones and the empty entries denote zeroes. We shall now generalize this idea to higher dimensions.

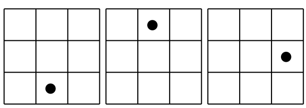
DEFINITION. A *d-dimensional dot array* is a d-dimensional $[n_0] \times [n_1] \times \cdots \times [n_{d-1}]$ array P where every position $\mathbf{x} = (x_0, x_1, \ldots, x_{d-1})$ may be either empty or dotted. We will identify P with the set of positions that have a dot in P.

We denote by $P[\mathbf{x}]$ the *principal subarray* of P consisting of all entries at positions componentwise less than or equal to $\mathbf{x}$. (So, for example, in the two-dimensional case the principal subarrays are the upper left submatrices.)

The space of positions can be given a partial order by $\mathbf{x}' \leq \mathbf{x}$ if $x_i' \leq x_i$ for all indices i, so that a principal subarray is a lower interval in the poset. The poset also has a *join* operation, defined by componentwise maximum: $\mathbf{x} \vee \mathbf{y} = \mathbf{z}$ where $z_i = \max(x_i, y_i)$ for all coordinate indices i.

For an arbitrary d-dimensional dot array P, say that the *rank of P along the j-axis*, denoted by $\mathrm{rk}_j P$, is the number of values of the index x_j such that there exists at least one dot in P in some position whose jth coordinate is x_j. If $\mathrm{rk}_j P = r$ for all $j = 0, \ldots, d-1$, so that the rank is the same along any axis, then we say that P is *rankable* with $\operatorname{rank} P = r$. The intuitive picture is that in whichever direction we traverse P, the number of layers containing a dot will be the same.

EXAMPLE. Let P be the dotted $[3]^3$-array shown below as three 3-by-3 layers. The zeroth coordinate is the row index (top to bottom), the first coordinate is the column index (left to right), and the second coordinate is the layer index (left to right).



This dot array is not rankable. On the one hand, we have $\mathrm{rk}_0 P = \mathrm{rk}_2 P = 3$; on the other hand, since there is no dot in the first column of any layer, we have $\mathrm{rk}_1 P = 2$.

DEFINITION. Say that two dot arrays P and P' of the same shape are *rank equivalent* if for every position $\mathbf{x}$ and every coordinate index i we have $\mathrm{rk}_i P[\mathbf{x}] = \mathrm{rk}_i P'[\mathbf{x}]$.

We shall investigate when we can remove a dot and stay in the rank equivalence class. For a position $\mathbf{x}$, let $P - \mathbf{x}$ and $P + \mathbf{x}$ denote the two dot arrays obtainable from P by removing the dot in position $\mathbf{x}$ (if there is one, otherwise do nothing) and adding a dot in position $\mathbf{x}$ (if there is none) respectively.

LEMMA 2.1. *For a dot array P and a position $\check{\mathbf{x}}$, the dot arrays $P - \check{\mathbf{x}}$ and $P + \check{\mathbf{x}}$ are rank equivalent if and only if $\check{\mathbf{x}} = \bigvee \mathscr{X}$ for some subset of dots $\mathscr{X} \subseteq P$ not containing $\check{\mathbf{x}}$.*

Proof. We want to study when the ranks are independent of whether or not there is a dot in $\check{\mathbf{x}}$. Clearly, for every coordinate index i, the rank along the i-axis is independent of whether there is a dot in $\check{\mathbf{x}}$ if and only if there exists some dot $\mathbf{x}^{(i)} < \check{\mathbf{x}}$ with $x_i^{(i)} = \check{x}_i$. Take $\mathscr{X} = \{\mathbf{x}^{(0)}, \dots, \mathbf{x}^{(d-1)}\}$ and the lemma follows. ∎

In the two-dimensional case, this means for example that adding a dot in a position that is later in the row than some other dot, and later in the column than yet another dot, alters neither the column rank nor the row rank; e.g.,

	•
•	•

$\sim$

	•
•	

.

DEFINITION. Let us say that P is *totally rankable* if every principal subarray of P is rankable. If P is totally rankable then we can define the *rank array* of P as the integer array (of the same shape as P) whose entry at position $\mathbf{x}$ is rank $P[\mathbf{x}]$.

EXAMPLE. Below, we show a totally rankable $[3]^3$-array and the corresponding rank array.

	•	

	•	
•		

		•

0	0	0
0	0	0
0	1	1

0	1	1
0	1	1
1	2	2

0	1	1
0	1	2
1	2	3

Different totally rankable dot arrays may be rank equivalent, that is, yield the same rank array. In Section 4, we shall characterize the rank equivalence classes and choose a canonical representative of each class.

3. A CHARACTERIZATION OF TOTALLY RANKABLE DOT ARRAYS

In the previous section we defined a rank function on dot arrays and found that not all dot arrays are totally rankable, i.e., have well-defined rank in every position. In the two-dimensional world of classical permutation matrices, total rankability is easily characterized:

OBSERVATION 3.1. *A two-dimensional dot array P is totally rankable if and only if the first dot in any row is also the first dot in its column, and vice versa.*

Proof. This is an easy exercise. ∎

The objective of the next few pages is to derive the following general characterization of total rankability.

THEOREM 3.2. *Let P be a dot array. The following conditions are equivalent*:

(1) *P is totally rankable.*

(2) *Every two-dimensional projection of every principal subarray of P is totally rankable.*

(3) *If there are two dots in P in positions $\mathbf{x}'$ and $\mathbf{x}''$ and two coordinate indices i and j such that $x'_i < x''_i$ and $x'_j = x''_j$, then there must exist a dot in some position $\mathbf{x} \le \mathbf{x}' \vee \mathbf{x}''$ such that $x_i = x''_i$ and $x_j < x''_j$.*

(4) *Every* redundant *position in P is* covered *by dots in P.*

Remark. When the dimension is one, then every dot array is totally rankable and (2), (3), and (4) are empty conditions. When the dimension is two, condition (3) is equivalent to Observation 3.1.

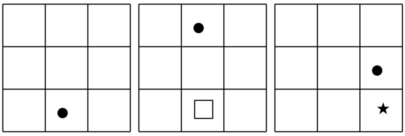
EXAMPLE. The *sparse* dot arrays, which we discussed in the Introduction, are totally rankable, since they trivially satisfy Condition 3 of Theorem 3.2.

We will begin by introducing the notions of redundant and covered positions, in order to proceed with projections, and finally the proof of the theorem.

DEFINITION. A position $\check{\mathbf{x}}$ is *redundant* in P if $\check{\mathbf{x}} = \bigvee \mathscr{X}$ for some $\mathscr{X} \subseteq P$ such that $|\mathscr{X}| \ge 2$ and every member in $\mathscr{X}$ has at least one coordinate in common with $\check{\mathbf{x}}$. In other words, a position is redundant if it can be written as the join of some positions of dots in P in a nontrivial way. A redundant position is *covered* if the set $\mathscr{X}$ above also satisfies:

1. $\check{\mathbf{x}} \notin \mathscr{X}$, and
2. for every j there is an $\mathbf{x} = (x_0, \ldots, x_{d-1})$ in $\mathscr{X}$ such that $x_j < \check{x}_j$.

EXAMPLE. Let us return to our previous example of a *not* totally rankable $[3]^3$-array. It has two redundant positions, one of which is not covered. Below we mark the covered position with a star and the other redundant position with a square.



Let $R(P)$ denote the set of redundant positions. The fundamental property of redundant positions is that it does not matter for the ranks if we place dots in them, as the next lemma shows.

LEMMA 3.3. *Given a dot array P and $\check{\mathbf{x}} \in R(P) \setminus P$ (i.e., a nondotted redundant position), the following must hold.*

(a) $P + \check{\mathbf{x}}$ *is rank equivalent to* P, *and*

(b) $R(P + \check{\mathbf{x}}) = R(P)$.

Proof. (a) This follows from Lemma 2.1. For Part (b), just observe that any member of $R(P + \check{\mathbf{x}})$ that is not in $R(P)$ would have to be of the form $\bigvee \mathscr{X}'$ where $\check{\mathbf{x}} \in \mathscr{X}'$. But since $\check{\mathbf{x}} = \bigvee \mathscr{X}$, we have that $\bigvee \mathscr{X}' = \bigvee(\mathscr{X} \cup \mathscr{X}' - \check{\mathbf{x}})$, and it is clear that the assumptions that $|\mathscr{X}'| \geq 2$, $|\mathscr{X}| \geq 2$, and $\check{\mathbf{x}} \in \mathscr{X}$ imply that $|\mathscr{X} \cup \mathscr{X}' - \check{\mathbf{x}}| \geq 2$. Hence $\bigvee \mathscr{X}'$ is a member also of $R(P)$. ∎

3.1 *Projections.* We will project dots to arrays of lower dimension, according to the following definition.

DEFINITION. Given a d-dimensional dot array P of shape $[n_0] \times \cdots \times [n_{d-1}]$ and k indices $i_0 < \cdots < i_{k-1}$, a k-dimensional *projection* P^π of shape $[n_{i_0}] \times \cdots \times [n_{i_{k-1}}]$ is induced by the map

$$\pi\colon (x_0, \ldots, x_{d-1}) \longmapsto (x_{i_0}, \ldots, x_{i_{k-1}}).$$

The dots of P are projected to P^π in the natural way: $P^\pi = \pi(P)$. There is a corresponding inclusion map $\iota\colon P^\pi \to P$ given by

$$\iota\colon (x_{i_0}, \ldots, x_{i_{k-1}}) \longmapsto (x_0, \ldots, x_{d-1}),$$

where x_i takes its maximal value n_i for every $i \notin \{i_0, \ldots, i_{k-1}\}$.

Projections are rank preserving in the following sense.

LEMMA 3.4. *Given a dot array P, indices $\{i_0, \ldots, i_{k-1}\}$, and the maps π and ι as above, the rank equality*

$$\mathrm{rk}_i P^\pi[\mathbf{x}] = \mathrm{rk}_i P[\iota(\mathbf{x})]$$

holds for any $i \in \{i_0, \ldots, i_{k-1}\}$. In particular, if P is totally rankable then P^π is totally rankable.

Proof. The projections of the dots that contribute to the rank along the i-axis in $P[\iota(\mathbf{x})]$ will similarly contribute to the rank along the i-axis in $P^\pi[\mathbf{x}]$, and vice versa. ∎

3.2. *Proof of the Characterization Theorem.* We shall now prove the characterization theorem by proving, in turn, that (1) ⇔ (2), (2) ⇔ (3), and (3) ⇔ (4).

Proof of Theorem 3.2. (1) ⇒ (2): If P is totally rankable, then every principal subarray $P[\mathbf{x}]$ is totally rankable. By Lemma 3.4, all projections of totally rankable arrays are totally rankable.

(2) ⇒ (1): If the two-dimensional projection P^π of $P[\mathbf{x}]$ induced by π: $(x_0, \dots, x_{d-1}) \mapsto (x_i, x_j)$ is totally rankable, then in particular $\mathrm{rk}_i P^\pi[(x_i, x_j)] = \mathrm{rk}_j P^\pi[(x_i, x_j)]$ and by Lemma 3.4 we therefore have $\mathrm{rk}_i P[\mathbf{x}] = \mathrm{rk}_j P[\mathbf{x}]$. This conclusion holds for all positions $\mathbf{x}$ and all indices i and j, so P is totally rankable.

(2) ⇔ (3): Condition (3) is clearly equivalent to saying that every two-dimensional projection of every principal subarray of P satisfies that the first dot in any row is also the first dot in its column, and vice versa. By Observation 3.1, this is equivalent to the projections in question being totally rankable.

(3) ⇒ (4): Assume that (4) is false, so that some redundant position $\check{\mathbf{x}} = (\check{x}_0, \dots, \check{x}_{d-1})$ is not covered. Since it is redundant, we can write $\check{\mathbf{x}}$ as $\bigvee \mathscr{X}$ for some set of dots $\mathscr{X} \subseteq P$ such that $|\mathscr{X}| \geq 2$ and for every $\mathbf{x}$ in $\mathscr{X}$ there is some k such that $x_k = \check{x}_k$. Since $\check{\mathbf{x}}$ is not covered, one of the two conditions from the definition of covered must be violated. This leaves open two cases; either

(a) it is impossible to write $\check{\mathbf{x}}$ as $\bigvee \mathscr{X}$ with $\mathscr{X} \subseteq P$ without having $\check{\mathbf{x}} \in \mathscr{X}$ (so there must be some coordinate index i such that $\check{\mathbf{x}}$ is the only dot satisfying $\mathbf{x} \leq \check{\mathbf{x}}$ and $x_i = \check{x}_i$); or

(b) there are some coordinate indices i and j such that that there is *no* dot $\mathbf{x} \leq \check{\mathbf{x}}$ with $x_i = \check{x}_i$ and $x_j < \check{x}_j$.

Choose any particular $\mathbf{x}' \neq \check{\mathbf{x}}$ in $\mathscr{X}$. In both cases (a) and (b) we can deduce that we can find coordinate indices i and j such that $x'_i < \check{x}_i$, $x'_j = \check{x}_j$, and such that there exists no dot $\mathbf{x} \leq \check{\mathbf{x}}$ with $x_j < \check{x}_j$ and $x_i = \check{x}_i$. Since $\check{\mathbf{x}} = \bigvee \mathscr{X}$, there must exist some $\mathbf{x}'' \in \mathscr{X}$ such that $x''_i = \check{x}_i$ (and hence $x''_j = \check{x}_j$). Thus (3) is false.

(4) ⇒ (3): Assume that every redundant position in P is covered by dots in P. We shall prove (3) for every principal subarray $P[\check{\mathbf{x}}]$ by induction on $\check{\mathbf{x}}$. Condition (3) holds trivially for the empty subarray. Let $P[\check{\mathbf{x}}]$ be a minimal unproven case and suppose that there are indices i and j such that (3) is false. Let $\mathscr{X}'$ and $\mathscr{X}''$ be the sets of dots that work as $\mathbf{x}'$ and $\mathbf{x}''$:

$$\mathscr{X}' = \{\mathbf{x}' \in P[\check{\mathbf{x}}] : x'_i < \check{x}_i, x'_j = \check{x}_j\}, \quad \mathscr{X}'' = \{\mathbf{x}'' \in P[\check{\mathbf{x}}] : x''_i = \check{x}_i, x''_j = \check{x}_j\}.$$

The assumption that (3) is false means that these sets are non-empty and that there is no dot in any position $\mathbf{x} \leq \check{\mathbf{x}}$ such that $x_i = x''_i$ and $x_j < x''_j$. The assumption that $P[\check{\mathbf{x}}]$ is the minimal unproven case implies that $\mathbf{x}' \vee \mathbf{x}'' = \check{\mathbf{x}}$ for every $\mathbf{x}' \in \mathscr{X}'$, $\mathbf{x}'' \in \mathscr{X}''$. In other words, for every coordinate index k, either $x'_k = \check{x}_k$ for every $\mathbf{x}' \in \mathscr{X}'$ or $x''_k = \check{x}_k$ for every $\mathbf{x}'' \in \mathscr{X}''$.

Since $\check{\mathbf{x}}$ is redundant it is also covered, which implies two things. First, in the set of dots covering $\check{\mathbf{x}}$ there must be some dot $\mathbf{x}'' \neq \check{\mathbf{x}}$ with $x''_i = \check{x}_i$. Hence there is a dot $\mathbf{x}''$ in $\mathscr{X}''$ with $x''_\ell < \check{x}_\ell$ for some index ℓ. Second, in the set covering $\check{\mathbf{x}}$ there must be some dot $\mathbf{z} \leq \check{\mathbf{x}}$ with $z_j < x''_j$ and hence $z_i < x''_i$, and with some coordinate in common with $\check{\mathbf{x}}$, say $z_k = \check{x}_k$. As we noted above, either $z_k = x'_k$ for every $\mathbf{x}' \in \mathscr{X}'$ or $z_k = x''_k$ for every $\mathbf{x}'' \in \mathscr{X}''$ (in which case $k \neq \ell$).

We now obtain a contradiction, because in the first case the dots $\mathbf{z}$ and $\mathbf{x}'$ and the coordinate indices j and k are such that the induction hypothesis (3) is applicable, yielding a dot $\mathbf{y} \leq \mathbf{x}' \vee \mathbf{z}$ (so $y_i < x''_i$) with $y_j = x'_j$ and $y_k < x'_k$. Then $\mathbf{y}$ is in $\mathscr{X}'$, contradicting the assumption of the first case. In the second case we have a contradiction by a similar argument with the dots $\mathbf{z}$ and $\mathbf{x}''$ and the indices k and i yielding a dot $\mathbf{y} \leq \mathbf{x}'' \vee \mathbf{z} < \check{x}$. The last strict inequality follows from $x''_\ell < \check{x}_\ell$ above and $z_\ell < \check{x}_\ell$ which we can assume since otherwise we would be in the first case.

Hence the assumption that (3) is false for $P[\check{\mathbf{x}}]$ must be wrong, and the claim follows by induction. ∎

4. PERMUTATION ARRAYS

As promised in the Introduction, we shall define permutation arrays as totally rankable dot arrays with minimal number of dots in the rank equivalence class. Hence, we must prove that if P is a totally rankable dot array, then there exists a unique minimal member of the rank equivalence class of P. In fact, we have a stronger result, thanks to the above characterization theorem.

4.1. *Structure of Rank Equivalence Classes.* We will show that rank equivalence classes of dot sets (arrays) are intervals in the boolean lattice on all possible dot sets.

PROPOSITION 4.1. *Two totally rankable dot arrays P and P' of the same shape are rank equivalent if and only if $P - R(P) \subseteq P' \subseteq P + R(P)$.*

First, we need to complement Lemma 3.3.

LEMMA 4.2. *Given a totally rankable dot array P and a redundant dot $\check{\mathbf{x}} \in P \cap R(P)$, the following must hold.*

(a) $P - \check{\mathbf{x}}$ *is rank equivalent to* P, *and*

(b) $R(P - \check{\mathbf{x}}) = R(P)$.

Proof. By the characterization theorem, the redundant position $\check{\mathbf{x}}$ is covered. Since $\check{\mathbf{x}}$ is covered, it is made redundant by a set $\mathscr{X}$ not containing $\check{\mathbf{x}}$, so $\check{\mathbf{x}}$ is redundant in $P - \check{\mathbf{x}}$. Now (a) and (b) follow from Lemma 3.3. ∎

Proof of Proposition 4.1. If P is totally rankable, Lemmas 3.3 and 4.2 combine to say that P' is rank equivalent to $P - R(P)$ as well as to $P + R(P)$ and to every set in the boolean interval between these two.

We must now show that no dot array P' outside this interval can be rank equivalent to P. Assuming the contrary, what would a minimal counterexample look like? Since rank equivalence is inherited by principal subarrays, it would be a pair P' and P differing only by a dot in P' in the maximal position $\mathbf{x}$, and such that $\mathbf{x} \notin R(P)$. But by Lemma 2.1, such a pair cannot be rank equivalent. ∎

4.2 *Permutation Arrays and Their Rank Arrays.* The result above allows us to finally define our notion of permutation arrays.

Definition. A *permutation array* of side length n and dimension d is a totally rankable dot array of shape $[n]^d$, with rank n and no redundant dots.

Example. We have already seen that sparse dot arrays are totally rankable. They obviously have rank n, and no redundant dots, so all sparse dot arrays are permutation arrays.

With this definition, it is clear that two-dimensional permutation arrays are equivalent to classical permutation matrices. Now we shall characterize what rank arrays of permutation arrays (or totally rankable arrays, since they are rank equivalent with permutation arrays) look like.

Remark. The following characterization is the key to the geometric application in [1]. We use it to show that the "relative position" of d complete flags $E_\bullet^{0}, \ldots, E_\bullet^{d-1}$, encoded by the $[n]^d$-array with entries

$$\dim\left(E^0_{x_0} \cap E^1_{x_1} \cap \cdots \cap E^{d-1}_{x_{d-1}}\right),$$

is always a rank array of a permutation array.

Proposition 4.3. *Given parameters n and d, let ρ be an $[n]^d$-array of nonnegative integers $\rho(\mathbf{x})$. Then ρ is the rank array of some permutation array P if and only if the three conditions below are satisfied. To simplify the conditions, we define $\rho(\mathbf{x}) = 0$ whenever any index x_k is 0.*

1. *The difference between two neighboring ranks,*

$$\rho(x_0, \ldots, x_k, \ldots, x_{d-1}) - \rho(x_0, \ldots, x_k - 1, \ldots, x_{d-1}),$$

equals 0 *or* 1 *for all* $\mathbf{x} \in [n]^d$.

2. *If*

$$\rho(x_0, \ldots, x_k, \ldots, x_{d-1}) - \rho(x_0, \ldots, x_k - 1, \ldots, x_{d-1}) = 1,$$

then

$$\rho(x'_0, \ldots, x'_k, \ldots, x'_{d-1}) - \rho(x'_0, \ldots, x'_k - 1, \ldots, x'_{d-1}) = 1,$$

whenever $x'_k = x_k$ *and* $\mathbf{x}' \geq \mathbf{x}$.

3. $\rho(n, n, \ldots, n)$, *the greatest rank, is* n.

Proof. Let us show that for the rank array of P the three conditions are satisfied.

(1) The difference between two neighboring ranks equals 1 or 0, depending on whether the differing layer contains a dot or not.

(2) If the difference is 1, i.e., if

$$\text{rank } P[x_0, \ldots, x_k, \ldots, x_{d-1}] - \text{rank } P[x_0, \ldots, x_k - 1, \ldots, x_{d-1}] = 1,$$

then there is a dot in the layer, and hence whenever $x'_k = x_k$ and $\mathbf{x}' \geq \mathbf{x}$ this dot will be counted also in the difference

$$\text{rank } P[x'_0, \ldots, x'_k, \ldots, x'_{d-1}] - \text{rank } P[x'_0, \ldots, x'_k - 1, \ldots, x'_{d-1}].$$

(3) By definition of permutation arrays, rank $P[n, n, \ldots, n] = n$.

For the other direction, given some ρ such that the three conditions hold, we must distribute dots in P such that ρ is the rank array of P. By (1) all differences between neighbors are either 0 or 1. By (2) all differences equaling 1 are determined by the minimal positions of such differences, that is, positions $\mathbf{x}$ such that

$$\rho(x_0, \ldots, x_k, \ldots, x_{d-1}) - \rho(x_0, \ldots, x_k - 1, \ldots, x_{d-1}) = 1$$
$$\text{for every } k = 0, 1, \ldots, d-1.$$

Hence we get a totally rankable array with the desired rank array just by putting a dot in every minimal position. By (3) this dot array has full rank, so it is rank equivalent to a permutation array. ∎

Example. Below, we show a 3-by-3-by-3 rank array and the totally rankable dot array that one obtains by following the proof and putting a dot in every position where the rank has increased in all directions. If the redundant dots at $(3, 2, 2)$ and $(3, 3, 3)$ are removed we retrieve the permutation array of a previous example.

0	0	0	0	**1**	1	0	1	1
0	0	0	0	1	1	0	1	**2**
0	**1**	1	**1**	**2**	2	1	2	**3**

				•				
								•
	•		•	•				•

5. GENERATION OF PERMUTATION ARRAYS

Permutation arrays can be generated in an efficient way. We will describe here a recursive algorithm for constructing every permutation array for specified n and d.

Let P be a totally rankable $[n]^d$-array of rank $k > 0$ with no redundant dots. Let A be an antichain of dots in P, that is, for every pair of dots $\mathbf{x}$ and $\mathbf{y}$ in A there are coordinate indices i and j such that $x_i < y_i$ and $x_j > y_j$. Let $\widetilde{A}$ be the set of positions that are covered by A.

DEFINITION. *Downsizing* of P with respect to the antichain A is done by removing from P the dots of the antichain A, adding the dots of $\widetilde{A}$, and finally removing all redundant dots. (It is called downsizing since it implies getting rid of some dots and replacing them with new and fewer dots at lower positions!) We say that the downsizing was *successful* if the resulting array is totally rankable of rank $k - 1$ (one lower than P).

PERMUTATION ARRAY CONSTRUCTION ALGORITHM. For dimension $d = 1$ there is a unique permutation array of side length n, namely the one where every position has a dot.

For $d \geq 2$ we give the following inductive algorithm to construct an $[n]^d$-permutation array P:

1. Choose an $[n]^{d-1}$-permutation array P_1 and set a counter $i = 1$.

2. Choose an antichain A_i of dots in P_i such that the downsizing of P_i with respect to A_i is successful. Let P_{i+1} be the resulting array.

3. Increase i by 1 and repeat from step 2 if $i < n$, otherwise let $A_n = P_n$.

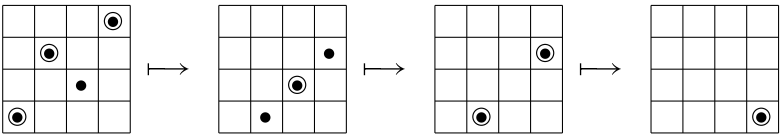
4. Let P be the $[n]^d$-array with dots

$$\{(x_0, \ldots, x_{d-2}, i) : (x_0, \ldots, x_{d-2}) \in A_{n+1-i}\},$$

that is, the antichains chosen in step 2 form the layers of P from bottom and up.

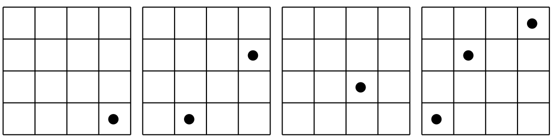
Remark. In the case of $d = 3$, there is a connection between this algorithm and an algorithm of Shapiro *et al.* [4] for generating certain "chains of permutations"-related pairwise intersections of Schubert cells. We discuss the connection to our geometric problem in [1].

EXAMPLE. Let P_1 be the totally rankable array to the left below, with the dots in an antichain A_1 circled. Then downsizing with respect to A_1 results in the array P_2 to the right, etc.



P_1 P_2 P_3 P_4

Putting the antichains as layers from the bottom up gives us the following permutation array:



THEOREM 5.1. *The output from the Construction Algorithm is a $[n]^d$-permutation array. Furthermore, all $[n]^d$-permutation arrays can be produced with the Construction Algorithm.*

The definitions of redundant and covered depend in a crucial way on the dimension of the array. To clarify the proof below we will qualify our terms by saying "d-redundant," "d-covered," etc. By the *bottom layer* of an $[n]^d$-array P we will mean the $(d-1)$-dimensional layer whose last coordinate is maximal: $x_{d-1} = n$. Let henceforth π be the projection map π: $(x_0, \ldots, x_{d-2}, x_{d-1}) \mapsto (x_0, \ldots, x_{d-2})$, i.e., the projection onto the bottom layer.

LEMMA 5.2. *Let $P' = \pi(P)$ be the $(d-1)$-dimensional projection of a d-dimensional dot array P onto its bottom layer and assume that the bottom layer of P is totally $(d-1)$-rankable (although P is not necessarily totally rankable). If $\mathbf{x}$ is a d-redundant position in the bottom layer of P, then either $\mathbf{x}$ is d-covered in P or its projection $\pi(\mathbf{x})$ is $(d-1)$-redundant in P'.*

Proof. If $\mathbf{x} = \bigvee \mathscr{X}$ in P, then $\pi(\mathbf{x}) = \bigvee \pi(\mathscr{X})$ in P'. If there were a dot $\mathbf{x}_1 \in \mathscr{X}$ that had only the last coordinate in common with $\mathbf{x}$, then $\pi(\mathbf{x}_1)$ cannot be used to show that $\pi(\mathbf{x})$ is redundant. We will still have $\pi(\mathbf{x}) = \bigvee(\pi(\mathscr{X})\backslash\{\pi(\mathbf{x}_1)\})$. Hence, $\pi(\mathbf{x})$ is $(d-1)$-redundant in P' unless $\mathscr{X} = \{\mathbf{x}_1, \mathbf{x}_2\}$ where $\mathbf{x}_1$ has only the last coordinate in common with $\mathbf{x}$ and $\mathbf{x}_2$ has the first $d-1$ coordinates in common with $\mathbf{x}$. Since the bottom layer is assumed to be totally $(d-1)$-rankable, $\mathbf{x}_2$ cannot be in the bottom layer because it would then be redundant but not covered in the bottom layer. Hence, $\mathbf{x}$ is then d-covered in P by $\mathbf{x}_1$ and $\mathbf{x}_2$. ∎

We also need to know how ranks are affected by downsizing.

LEMMA 5.3. *Let T be the successful downsizing with respect to an antichain A of a totally rankable dotted $[n]^d$-array P (with no redundant dots). Then*

$$\operatorname{rank} P[\mathbf{x}] - \operatorname{rank} T[\mathbf{x}] = \begin{cases} 1 & \text{if } \mathbf{x} \geq \mathbf{y} \text{ for some } \mathbf{y} \in A, \\ 0 & \text{otherwise}. \end{cases}$$

Proof. Clearly, the rank in position $\mathbf{x}$ is not affected if no dot $\mathbf{y} \leq \mathbf{x}$ is removed or added. Otherwise, let $B \subset A$ be the subset of dots $\mathbf{y}$ in A such that $\mathbf{x} \geq \mathbf{y}$. Let $\widetilde{B} \subset \widetilde{A}$ be the similarly defined subset of $\widetilde{A}$. The downsizing will affect the rank in $\mathbf{x}$ by removing B and adding $\widetilde{B}$. First note that if $k_i := \min\{x_i : \mathbf{x} \in B\}$, then all the dots in $P[\mathbf{x}]$ with ith coordinate k_i have to be in B. If not, we would have $\operatorname{rank} P[(x_0, \ldots, x_{i-1}, k_i, x_{i+1}, \ldots, x_{d-1})] = \operatorname{rank} T[(x_0, \ldots, x_{i-1}, k_i, x_{i+1}, \ldots, x_{d-1})]$, which is a contradiction since we assumed that P had no redundant dots. This implies that $\operatorname{rank} P[\mathbf{x}] \leq \operatorname{rank} T[\mathbf{x}] - 1$. To prove equality assume that $\operatorname{rank} P[\mathbf{x}] \leq \operatorname{rank} T[\mathbf{x}] - 2$. Hence, in each coordinate direction at least two layers would have been emptied of dots by the downsizing. Let $y_0 < y_0'$ be the layers in the 0th coordinate direction and let $\mathbf{y}, \mathbf{y}' \in B$ be the corresponding dots $\mathbf{x} \geq \mathbf{y}, \mathbf{y}'$. Since no new dot with 0th coordinate y_0' has been formed as a cover, we conclude that $\mathbf{y}, \mathbf{y}'$ must have a coordinate in common, say coordinate i. Now we use the third statement in Theorem 3.2 on the coordinates 0 and i which implies that we must have yet another dot $\mathbf{y}''$ with coordinate y_0', but with a smaller ith coordinate and hence $\mathbf{y}'' \in \mathbf{B}$. If all three of $\mathbf{y}, \mathbf{y}', \mathbf{y}''$ have a common coordinate we apply Theorem 3.2 again to get yet another dot and so on until we have a collection of dots in $P[\mathbf{x}]$ which covers $\mathbf{y} \vee \mathbf{y}'$, a contradiction. ∎

Proof of Theorem 5.1. Let P be a dot array produced with the algorithm and note that the projection of P to the bottom layer is rank equivalent to the n^{d-1}-permutation P_1 chosen in Step 1.

In order to prove that P is totally rankable it suffices by Theorem 3.2 to prove that all d-redundant positions in P are also d-covered. Let $\mathbf{x}$ be a d-redundant position in the bottom layer of P. By Lemma 5.2, either $\mathbf{x}$ is

d-covered in P or $\pi(\mathbf{x})$ is $(d-1)$-redundant in P_1. In the latter case $\mathbf{x}$ is $(d-1)$-covered in P_1. Since $\mathbf{x}$ is d-redundant in P and lies in the bottom layer of P consisting of the antichain A, the position $\mathbf{x}$ must be larger than some element in A_1. Hence, $\pi(\mathbf{x})$ is $(d-1)$-covered by $B \cup C$, where $\emptyset \neq B \subset A_1$ and $C \subset P_1 \setminus A_1$. Then $\mathbf{x}$ is d-covered by B in the bottom layer and $C \cup \widetilde{B}$ in P_2, where $\widetilde{B}$ is the set of positions $(d-1)$-covered by B.

A position that is covered by A_1 and by dots in P_2 one layer up will still be covered when the dots in P_2 are spread out over different layers. We can use exactly the same argument for every layer, so we have now proved that the array P constructed by the algorithm is totally rankable. In order to conclude that P is a permutation array, we must also note that P has full rank (which is obvious since there are dots in every layer) and that P has no redundant dots. To see this last part, just note that if there were a dot in an antichain that is redundant after the downsizing it would have also been redundant before the downsizing.

We now proceed to the second statement of Theorem 5.1, which says that every permutation array can be constructed by the algorithm. Let P be a $[n]^d$-permutation. First we observe that in an $[n]^d$-permutation (which has no redundant dots by definition) every $(d-1)$-layer is an antichain. This is true since if we had two dots in a layer that were comparable then the larger one would be d-redundant. We now claim that we get P by choosing the projection $\pi(P)$ as P_1 in Step 1 and the antichains that make up the layers of P as the A_i in Step 2. Again it suffices to study the bottom-layer case. Let A_1 be the antichain in the bottom layer of P and let P_2 be the corresponding downsizing of P_1. We need to prove that P_2 is rank equivalent to $\pi(P \setminus A_1)$, the projection of P after its bottom layer is removed, which in turn is rank equivalent to layer $n-1$ of P, thanks to Lemma 3.4.

Let $\mathbf{x} = (x_0, \ldots, x_{d-2})$ be a generic $(d-1)$-position. We want to prove that $\operatorname{rank} P[(\mathbf{x}, n-1)] = \operatorname{rank} P_2[\mathbf{x}]$. Since the bottom layer of P contains only the antichain A, the rank between the bottom layer and the one above differs only for positions that are lower than some dot in A:

$$\operatorname{rank} P[\mathbf{x}, n] - \operatorname{rank} P[\mathbf{x}, n-1] = \begin{cases} 1 & \text{if } \mathbf{x} \geq \mathbf{y} \text{ for some } \mathbf{y} \in A \\ 0 & \text{otherwise.} \end{cases}$$

But by Lemma 5.3, we also have

$$\operatorname{rank} P_1[\mathbf{x}] - \operatorname{rank} P_2[\mathbf{x}] = \begin{cases} 1 & \text{if } \mathbf{x} \geq \mathbf{y} \text{ for some } \mathbf{y} \in A \\ 0 & \text{otherwise.} \end{cases}$$

Clearly the desired equality $\operatorname{rank} P[\mathbf{x}, n-1] = \operatorname{rank} P_2[\mathbf{x}]$ follows from the two equations above and the fact (Lemma 3.4 again) that $\operatorname{rank} P[\mathbf{x}, n] = \operatorname{rank} P_1[\mathbf{x}]$. ∎

See Fig. 1 for the 70 permutation arrays when $n = d = 3$.

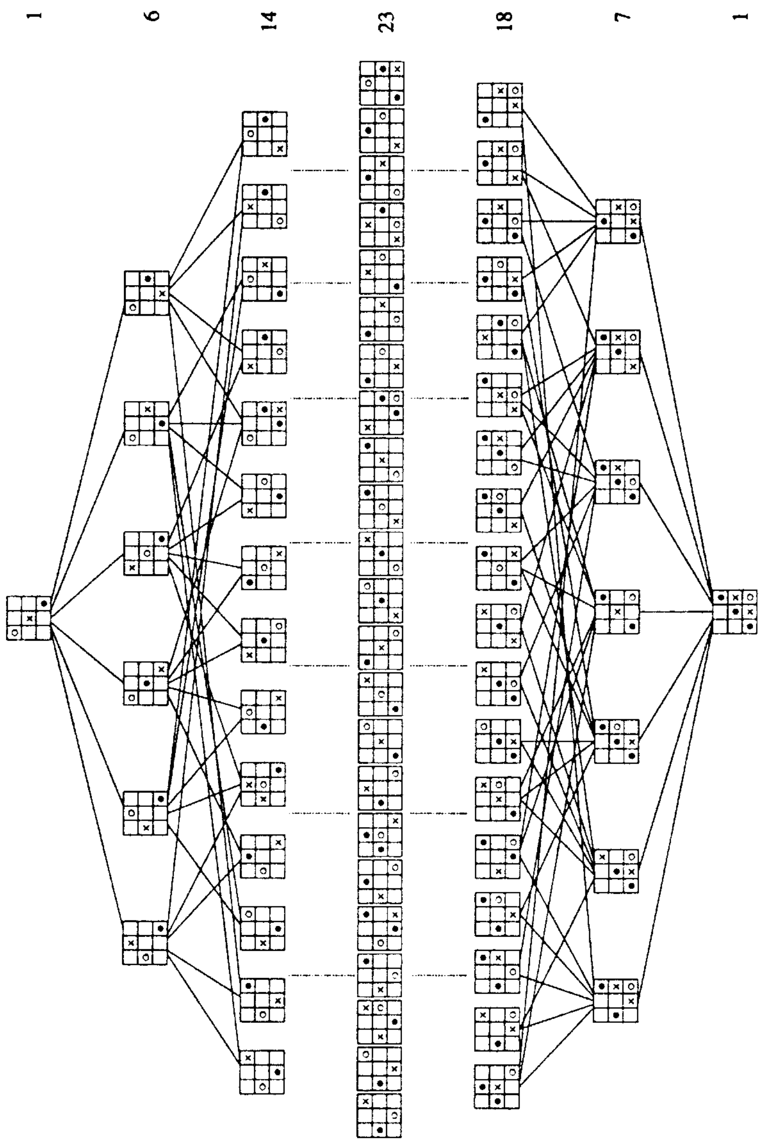


FIG. 1. The poset on $\mathscr{P}_{3,3}$ of $(3 \times 3 \times 3)$-permutation arrays. The empty circles, crosses, and filled circles signify dots in the first, second, and third layers, respectively. The edges in the middle were too numerous to be conveniently drawn.

6. ENUMERATION

Let $\mathcal{P}_{n,d}$ be the set of permutation arrays of side length n and dimension d. Let $p(n,d) = |\mathcal{P}_{n,d}|$ be the total number of permutation arrays. By using the construction algorithm of the previous section, we have been able to compute the following table.

From Table I, a combinatorialist easily recognizes that

$$p(n,2) = n! \qquad \text{and} \qquad p(2,d) = B(d),$$

where $B(d)$ is the dth Bell number. The first equation is obvious; indeed, the permutation arrays are defined to be generalizations of classical permutation matrices and specialize to them for $d = 2$. The second equation is implied by the following poset results, which we prove in [1].

Define the partial order $\geq_r$ on $\mathcal{P}_{n,d}$ by $P' \geq_r P$ if $\operatorname{rank} P'[\mathbf{x}] \geq \operatorname{rank} P[\mathbf{x}]$ for all positions $\mathbf{x}$. Then for $n = 2$, the poset is isomorphic to the partition lattice on an n-set! On the other hand, for $d = 2$ the poset coincides with the Bruhat order on S_n (in fact, it is well-known and is sometimes taken as the definition of Bruhat order).

6.1. *Bounds on* $p(n,d)$.

Explicit expressions for $p(n,d)$ are not known for $n \neq 2$, $d \neq 2$. A general lower bound is obtained from the fact that the $(n!)^{d-1}$ sparse $[n]^d$-arrays are permutation arrays, so that

$$(n!)^{d-1} \leq p(n,d)$$

We see that these numbers grow very fast. Upper bounds can, in principle, be obtained by an analysis of the construction algorithm, but this gets messy for higher values of d. For $d = 3$ we have the following.

PROPOSITION 6.1. *The number of three-dimensional permutation arrays has the upper bound*

$$p(n,3) \leq n! \cdot 2^{\binom{n+1}{2}-1}.$$

TABLE I
Table of Values of $p(n,d)$, the Number of $[n]^d$-Permutation Arrays

$n\backslash d$:	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	1
2	1	2	5	15	52	203	877	4140
3	1	6	70	1574	*69874			
4	1	24	2167	*968162				
5	1	120	130708					
6	1	720	14231289					
7	1	5040	2664334184					
8	1	40320	831478035698					

Note. Entries marked * have been computed with a program written by A. Ingestedt.

Proof. The proof relies on the construction algorithm from Section 5. The number of bottom layers is equal to the number of two-dimensional permutations $n!$. At level i there are i dots and hence at most $2^i - 1$ possible ways to downsize to level $i-1$, for $n \geq i \geq 2$. This proves the desired bound. ∎

It remains a challenge to find a nontrivial bound for the number of ways to do the downsizing step.

7. CONCLUSIONS

We have defined a class of hypercubic dot arrays that generalize permutation matrices in a certain sense. The motivation came from algebraic geometry; the permutation arrays encode intersections of d complete flags.

Despite the facts that permutation arrays have a fairly simple combinatorial definition, that they specialize to the well-known concepts of permutations for $d = 2$ and to set partitions for $n = 2$, and that they can be generated in a reasonably straightforward way, the general enumeration problem seems formidably difficult. Perhaps there exists a fruitful interpretation of permutation arrays in terms of both permutations and set partitions, although we have not seen it.

REFERENCES

1. K. Eriksson and S. Linusson, A decomposition of $Fl(n)^d$ indexed by permutation arrays, *Adv. Appl. Math.* **25** (2000), 212–227.
2. W. Fulton, "Young Tableaux with Applications to Representation Theory and Geometry," Cambridge University Press, Cambridge, UK, 1996.
3. E. Pascal, "*Die Determinanten*," Teubner-Verlag, Leipzig, 1900.
4. B. Shapiro, M. Shapiro, and A. Vainshtein, On combinatorics and topology of pairwise intersections of Schubert cells in $SL_n/\mathcal{B}$ *in* "Arnold–Gelfand Seminars," pp. 397–437. Birkhauser, Basel, 1997.